"Embedded Linux distribution for GEAM6425/GEA4LAN"

User's manual

Version: 3.0

March 4, 2011

About Evidence S.r.l.

Evidence is a company operating in the field of software for embedded real-time systems. It started in 2002 as a spin-off company of the Real-Time Systems (ReTiS) Lab of the Scuola Superiore Sant'Anna (Pisa, Italy). Today, Evidence is a dynamic company having collaborations in the field of electronics, telecommunications, automotives, and industrial automation.

People at Evidence are experts in the domain of embedded and real-time systems, with a deep knowledge on the design and specification flow of embedded software, especially for the embedded market.

Besides providing consultancy services, Evidence also provides: BSPs based on Linux for embedded devices, evaluation boards featuring most innovative 8, 16 and 32-bit microcontrollers for the embedded market, development tools for making embedded software development easier, and tools for the schedulability analysis of real-time tasks running on your final product.

Contact Info

Evidence Srl,
Via Carducci 56
Localitá Ghezzano
56010 S.Giuliano Terme
PISA Italy

Tel: +39 050 99 11 224
Fax: +39 050 99 10 812

For more information about Evidence products, please send an e-mail to the following address: info@evidence.eu.com. Other information about the Evidence product line can be found at the Evidence web site at: http://www.evidence.eu.com.

Contents

List of Figures

About this document

This document contains the instructions to compile the Evelin BSP for the GEAM6425/GEA4LAN boards. Evelin (which stands for "Evidence Embedded Linux") is the Linux distribution for embedded systems created, provided and maintained by Evidence Srl.

The purpose of this document is to provide information about the usage of the environment to developers who are going to use it.

Acronyms

BSP	Board Support Package
SDK	Software Development Kit

Description of the software

The Evelin BSP provided by Evidence contains:

- Build scripts for automatic generation of initial firmware images.
- The Das U-Boot [2] bootloader. Das U-Boot is an Open Source "universal" cross-platform bootloader supporting hundreds of embedded boards and a wide variety of CPUs (the one of GEA included).
- The Linux kernel 2.6.31 with JFFS2 support and drivers for the GEAM6425/GEA4LAN boards.
- JFFS2 filesystem containing:
  - Busybox [1] with network support.
  - Modbus communication protocol for automation
  - GNU Compiler Collection for C, C++, Objective-C, Objective-C++, Fortran and Java
  - ScicosLab a GTK version of Scilab

  - QT graphical libraries
  - Tslib library for the touchscreen support
  - Boa webserver
  - FTP and Telnet servers
  - Dropbear for SSH connections
  - Network utilities like tc, iproute and tcng for traffic shaping
  - Python support
  - Gambas support
  - SMTP mail client
  - Samba client
  - Mplayer for multimedia applications
  - lrzsz Unix communication package for XMODEM, YMODEM and ZMODEM file transfer protocols
  - dpkg, apt and aptitude support for handling of packages
  - Postgres SQL database
  - VNC and LXDE

- Around 1.5 GB of Debian packages that can be chosen at compile-time or installed at run-time.

- Complete source code.

Notice that the Evelin BSP is meant to be compiled only using the Evelin SDK [3] environment provided by Evidence.

1 Build

In this chapter we will build a Linux distribution for the GEAM6425 board. The same steps can be repeated for the GEA4LAN board with few differences. To build the Linux distribution for the GEAM6425 board, follow the next steps:

1. Install Evelin SDK on your Linux distribution, or run the virtual machine containing the development environment. Please refer to the *Evelin SDK User's manual* [3] for further information.

2. Enter the directory containing the source code: `cd ~/ev-sdk/workspace/evelin-bsp`

3. **Put into the `binaries/` directory any additional file you want to be present in the filesystem**

4. Configure the system:
   `sb2 -t imx25 make menuconfig`
   The screenshot shown in Figure 1.1 will appear.

5. Select the *Vendor/Product Selection* menu. Use the following settings:
   `Vendor:  imx25`
   `Board:  GEAM6425`
   as shown in Figure 1.2.

6. Select the *Kernel Configuration* menu. Use the following settings:
   `(Default) Kernel default selections`
   `[ ] Apply PREEMPT_RT patch`
   as shown in Figure 1.3 and Figure 1.4.

7. If you want to use the ccache option during compilation select the *Compiler cache (ccache) Selection* and check:
   `[*] Use ccache`
   as show in Figure 1.5.

8. Do not change the settings for the *Toolchain Path Selection* option.

9. Select *Exit* and save the current settings.

10. The system will build the next menu. Select the *Simple System* option as show in Figure 1.6.
    i

11. Select the *(Default) Busybox* menu. Use the following settings:
    `(X) Default`
    `( ) Custom`
    as shown in Figure 1.7.

12. Select *Graphics* and choose the packages needed, as shown in Figure 1.8. Note that you need both *QT Libraries* and *Touchscreen* to run graphical applications.

13. Exit from the menuconfig environment. Then type:
    `sb2 -t imx25 make`

14. At the end of the compilation, the `images/` directory will contain the following files:
    - `u-boot.bin`: binary image of the U-Boot bootloader
    - `uImage`: binary image of the Linux kernel (in U-Boot format)
    - `filesystem.jffs2`: Linux filesystem (in JFFS2 format)

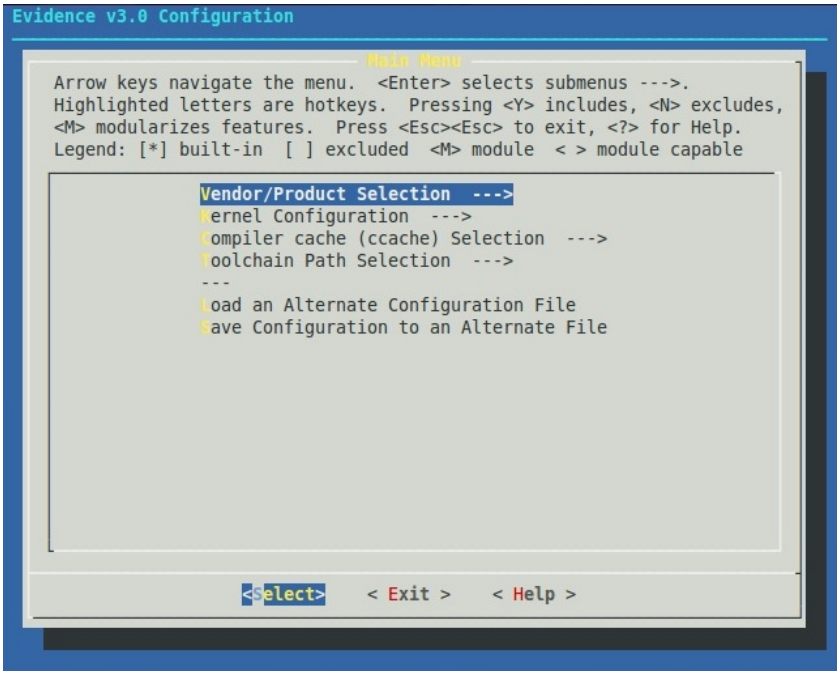

Figure 1.1: Screenshot of `make menuconfig`.

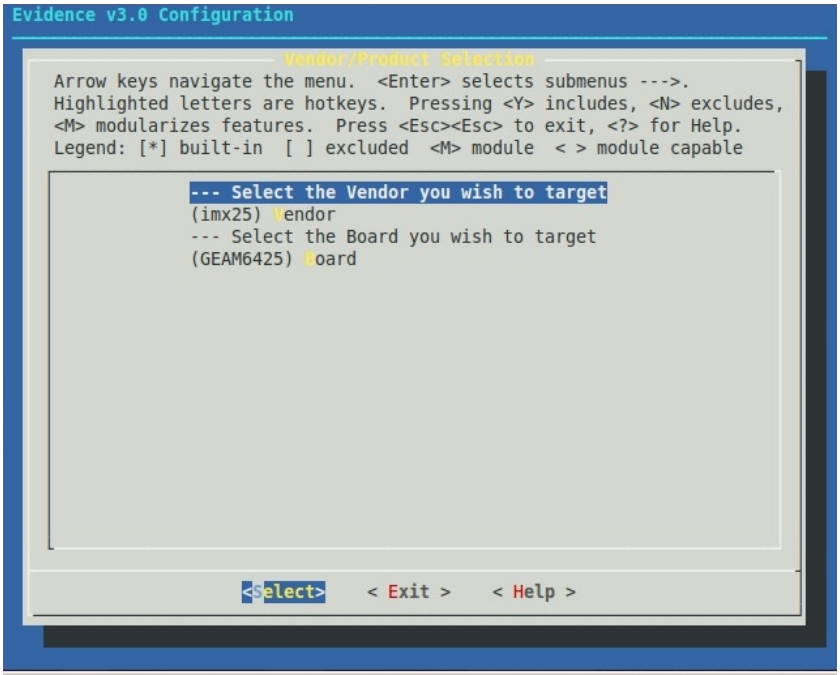

Figure 1.2: Target selection.

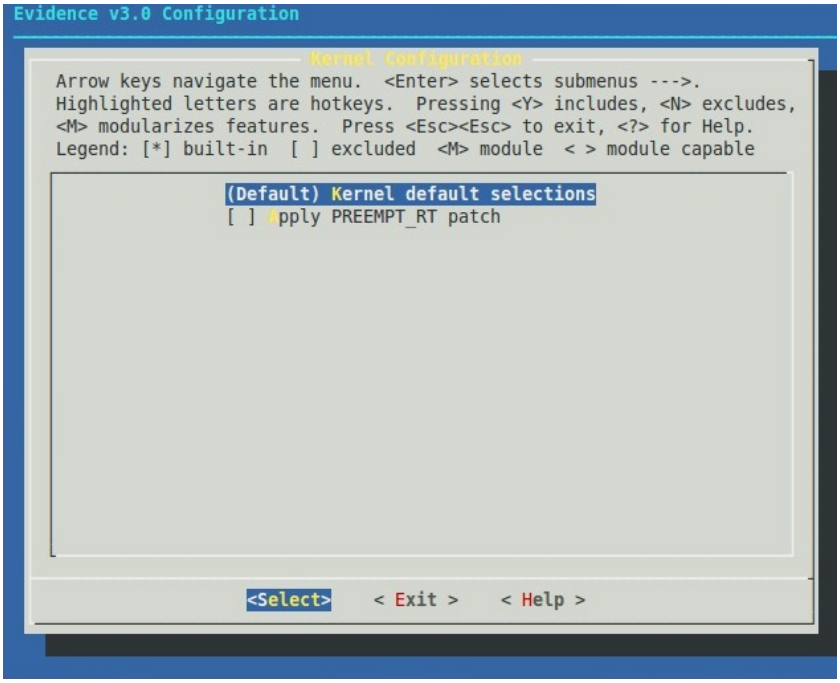

Figure 1.3: Kernel patch selection.

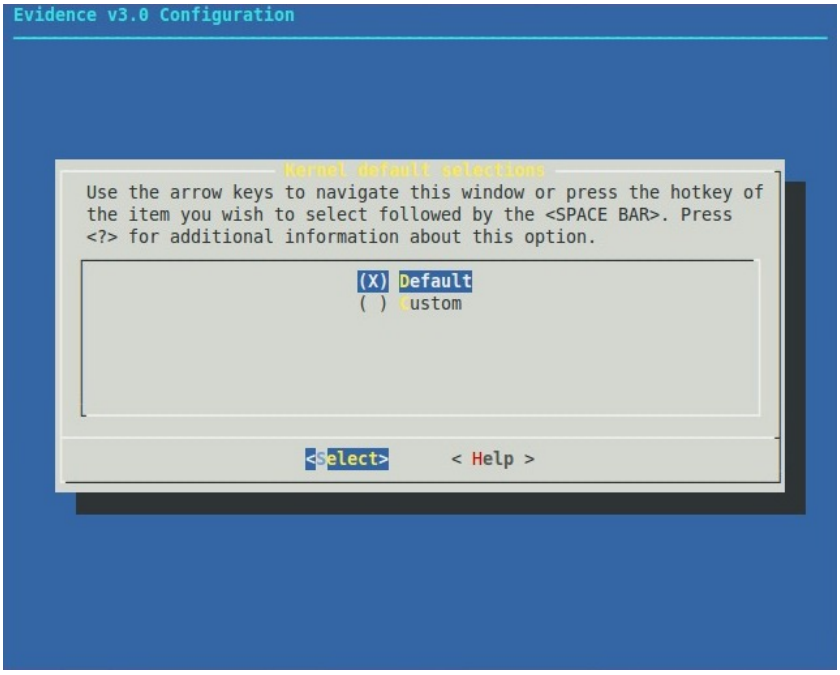

Figure 1.4: Kernel configuration.

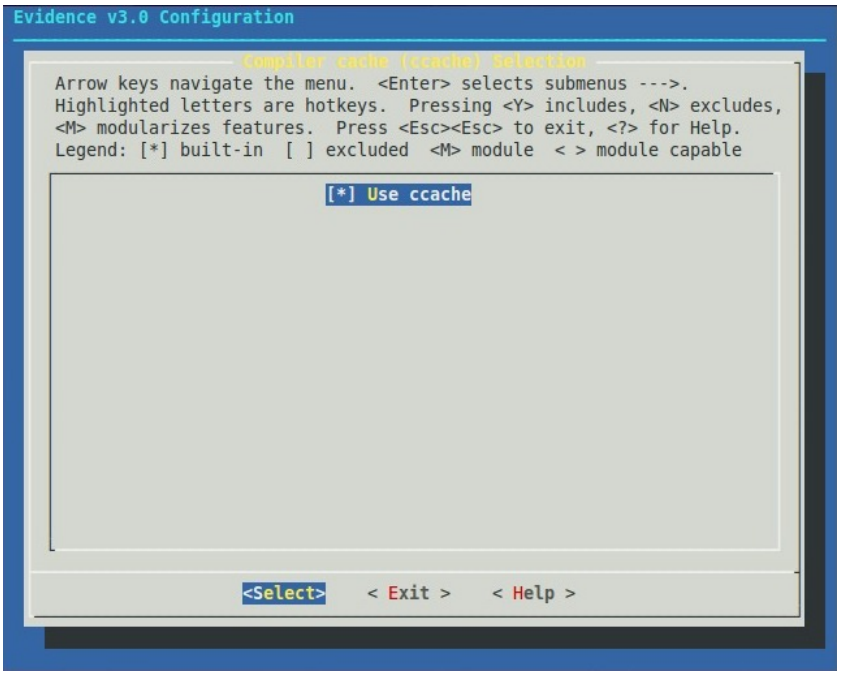

Figure 1.5: ccache selection.

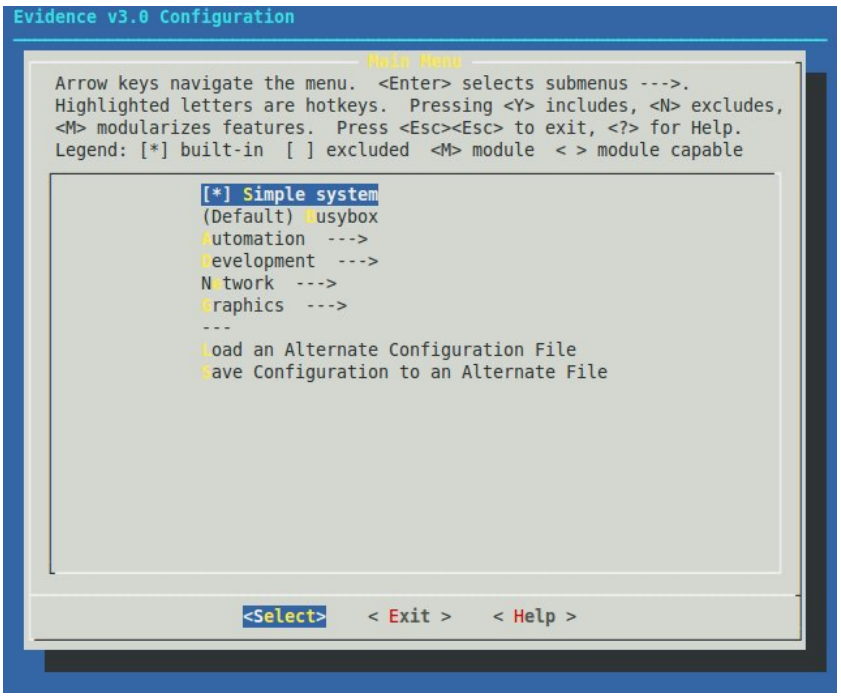

Figure 1.6: Busybox and filesystem selection.

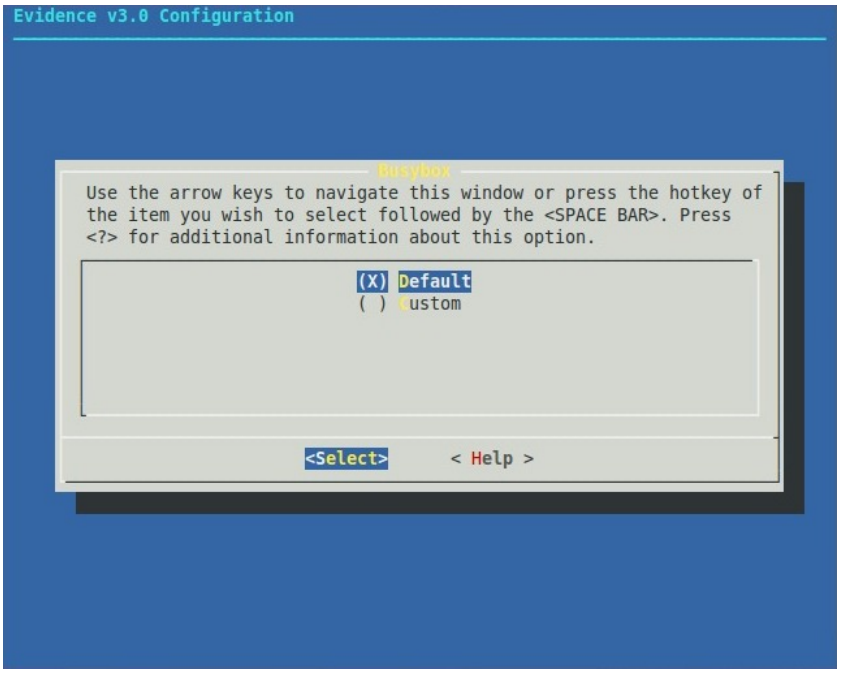

Figure 1.7: Busybox configuration.

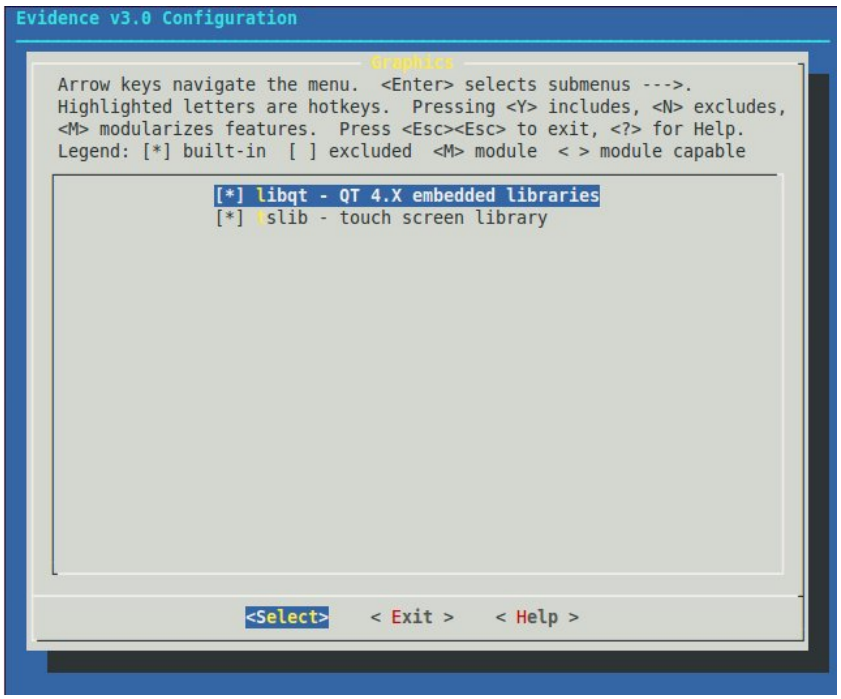

Figure 1.8: Packages selection.

1.1 Create a custom filesystem

To add some external files into target filesystem you can put them into `$HOME/ev-sdk/workspace/evelin-bsp/binaries` directory and type:

```
sb2 -t imx25 make create_filesystem
```

The new file filesystem.jffs2 in *images* directory will contain the additional files inserted.

IMPORTANT: manually modifying the *filesystem/romfs* directory could produce a bad filesystem image. Please do not add or remove files from this directory.

2 Connecting to the device

2.1 Serial connection

To open a console on the GEAM6425 board (which does not have a keyboard) you need use a remote serial communication program running on a host machine.

To connect to the device, follow the next steps:

1. Connect a serial cable to the serial interface of the target and to your PC
2. Start your favorite terminal tool (e.g., Hyperterminal, Putty, Minicom, Kermit) and set the following parameters:

Speed (bps)	115200
Parity	None
Data Bits	8
Stop Bits	1
Flow Control	Xon/Xoff

2.2 File transfer

The target runs a minimal FTP server for file transfers. To transfer files (e.g., applications) to the target, follow the next steps:

1. Connect to the target as explained in Section 2.1. Then, on the target, run the `ifconfig` tool to know the IP address of the device
2. On a PC, run your favorite FTP client to connect to the device (whose IP address has been discovered in the previous step). For instance, run:
   `ftp` *`target_ip_address`*

Note: the FTP server does not support authentication, thus there is no need of id and passwords.

2.3 TFTP transfer

If you run a VMware machine you need to setup the TFTP server for export its service on the network. For do this, follow the next steps:

1. Open VMware Player and select the evelin-bsp machine image.
2. On menu select "Virtual Machine - Virtual Machine Settings" as shown in Figure 2.1.
3. For Network Adapter select "Bridged: Connected directly to the physical network" and press Save as show in Figure 2.2.
4. Login into VMware machine and get the IP address of your Ethernet interface (if DHCP isn't present on network set your own IP address with `ifconfig` command).
5. The IP address of the VMware machine is the `serverip` value to set in target's U-Boot.

On VMware machine the default location of TFTP directory is `/tftp_boot`. The files generated by the compile process have to be copied into this directory.

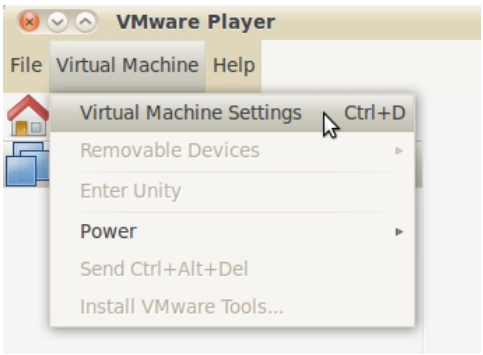

Figure 2.1: VMware settings menu.

2.4 NFS

To setup a NFS on your VMware machine, follow the next steps:

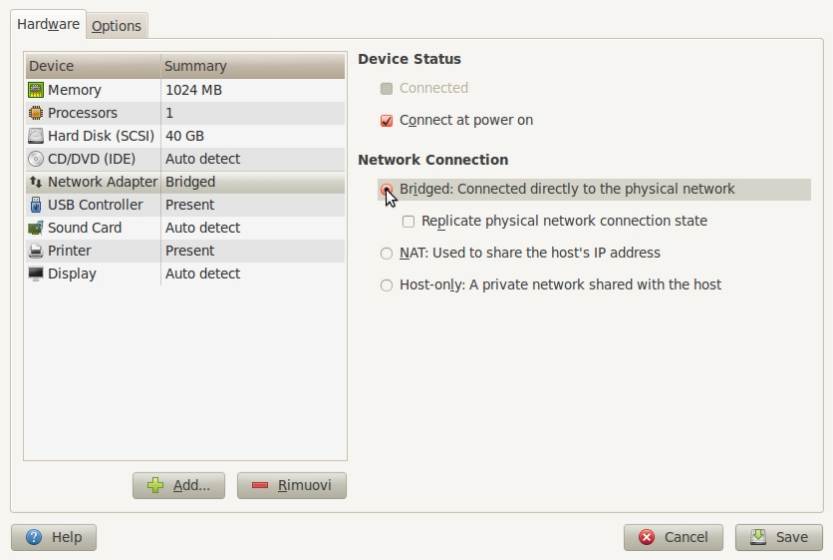

Figure 2.2: VMware network configuration menu.

1. Open a terminal console and type:
   `sudo vi /etc/exports`

2. Add a line at the end of file like:
   `/nfs_gea <network address>/<network mask>(rw,sync,no_root_squash)`
   where:

   - *network address* is the address of your local network.
   - *network mask* is the mask of your local network.

3 Flashing

On the NAND of the board, binaries are located at the following addresses:

Address	Size	Content
0x0000000	0x00FFFFF	`u-boot.bin`
0x0100000	0x02FFFFF	`uImage`
0x0400000	0x3BFFFFF	`filesystem.jffs2`

To flash the system, follow the next steps:

1. Compile the firmware, as explained in Chapter 1
2. Put the images created in the `images/` directory on a working TFTP server
3. Connect a serial cable to the target, and start a terminal program as explained in Chapter 2.
4. Reboot the board
5. At the U-Boot prompt, press a key to stop the automatic boot sequence.
6. On the U-Boot prompt, set the TFTP IP address using the following command:
   `set serverip <IP address>`
7. On the U-Boot prompt, set a static IP address for the board through the following command:
   `set ipaddr <IP address>`
8. Transfer the U-boot image and flash it through the following commands [1]:
   ```
   nand erase 000000 FFFFF
   tftp u-boot.bin
   nand write 0x81000000 00000 100000
   ```

[1] If U-Boot does not start anymore, or you have to flash it for the first time, see Section 4.1.

9. Transfer the Linux kernel and flash it through the following commands:
   ```
   nand erase 100000 300000
   tftp uImage
   nand write 0x81000000 100000 300000
   ```

10. Transfer the filesystem and flash it through the following commands:
    ```
    nand erase 400000 F7FFFFF
    tftp filesystem.jffs2
    nand write 0x81000000 400000 ${filesize}
    ```

11. Reboot the board

4 Usage of ATK

ATK is a utility running on Microsoft Windows and developed by Freescale to flash images on the board using a USB cable. Notice that this utility has been developed by Freescale (not by Evidence), hence you have to refer to Freescale for any technical issue.

Currently, ATK does not support the Microsoft Windows 7 64-bit operating system.

ATK can be run also on Linux through the `wine` emulator. To run ATK, you need to install wine and create a link to the serial console in the `.wine/dosdevices/` directory (e.g., `ln -s /dev/ttyUSB0 com1`).

4.1 How to use ATK to restore U-Boot

In case U-Boot does not start for some reason, you need to flash it using the i.MX Advanced Tool Kit (ATK) located in the `atk/` directory. ATK is a Freescale software working only under Windows XP OS.

To flash U-Boot, use the following procedure:

1. Insert the jumper JM2 in the geaM6425 board
2. Install ATK on Windows using the `FSL_ATK_TOOL_WINS_STD_INSTALL_xxx.exe` file
3. Connect the USB OTG device of the board to the USB interface of the PC
4. Start ATK on Windows
5. Select the following values:
   - *IMX CPU* `i.MX25_T01.1` architecture
   - *Device Memory Initial* `DDR2` memory type
   - *Host Setting* `USB` port
6. Press *Next*, select *Flash Tool* and than press *Go*

7. In the flash tool utility select:
   - *Flash Model* `Custom Model` and provide the `mx25_nand.bin` file
   - *Operation Type* `Erase`
   - *Address* `0x0000000`
   - *Size* `0x00FFFFF`
8. Press *Erase* on the right side of the window
9. In the flash tool utility select:
   - *Flash Model* `Custom Model` and provide the `mx25_nand.bin` file
   - *Operation Type* `Program` and check `read back check` option.
   - *Address* `0x00000000`
   - *Image File* provide the `uboot.bin` file
10. Press *Program* on the right side of the window
11. Wait until the entire process finish and check if *Flash program successful* is reported.
12. Remove the jumper JM2 from the board
13. Restart the board

Once the board is restarted enter in the u-boot menu (typing ENTER on boot) and insert the following parameters:

- `setenv fec_addr 00:00:00:00:00:01`
- `setenv serverip <IP address of TFTP server>`
- `setenv ipaddr <IP address of board`
- `setenv ethprime fec`
- `setenv ethaddr 00:50:C2:A0:E0:3E`
- `setenv netdev eth0`

- for 3.5" board

  ```
  setenv bootargs console=ttymxc0,115200 root=/dev/mtdblock2
  rootfstype=jffs2 video=mxcfb:320x240,16bpp,Var-QVGA
  ```

  for 7" board

  ```
  setenv bootargs console=ttymxc0,115200 root=/dev/mtdblock2
  rootfstype=jffs2 video=mxcfb:800x480,16bpp,Amp-WD
  ```

- `save`
- `reset`

4.2 How to use ATK to flash Linux and filesystem

With ATK is also possible to flash *uImage* and *filesystem.jffs2* images using the appropriate addresses specified in Chapter 3 (i.e., 0x0100000 for uImage and 0x0400000 for filesystem.jffs2).

Important: remember to insert the jumper and to use the *Erase* command before using the *Program* command otherwise the NAND will not be flashed correctly. If you want to erase the whole flash, just specify `Address: 0x00000000` and `Size:0x10000000`.

5 Further commands

5.1 Networking

5.1.1 MAC address

To set the MAC address on U-Boot type:

`setenv fec_addr <MAC address>`

For instance, to set the MAC address equal to 00:00:00:00:00:01, type:

`setenv fec_addr 00:00:00:00:00:01`

5.1.2 NFS mount

To mount a directory available on a NFS server on a target's local directory type on Linux:

`mount -t nfs -o nfsvers=3,nolock <server IP>:<server directory> <target directory>`

5.2 Users

To add a new user on Linux, type:

`adduser <name>`

This will automatically create the `/etc/shadow` file. Use the command

`passwd <name>`

to set the password.

5.3 Linux kernel

5.3.1 External kernel modules

To build a kernel module located in a directory different than the directory containing the Linux kernel, enter the directory containing the module and type:

```
sb2 -t imx25 make -C $HOME/ev-sdk/workspace/evelin-bsp/linux/linux-2.6.31-geaM6425
M=`pwd` modules
```

5.3.2 tc and tcng support

To use the `tc` and `tcng` tools, the Linux kernel must be compiled with the following options enabled in the `.config` file:

```
CONFIG_NETFILTER=y
CONFIG_NETFILTER_ADVANCED=y
CONFIG_NETFILTER_NETLINK=y
CONFIG_NETFILTER_NETLINK_QUEUE=y
CONFIG_NET_SCHED=y
CONFIG_NET_SCH_CBQ=y
CONFIG_NET_SCH_HTB=y
CONFIG_NET_SCH_PRIO=y
CONFIG_NET_SCH_SFQ=y
CONFIG_NET_SCH_DSMARK=y
CONFIG_NET_SCH_INGRESS=y
CONFIG_NET_CLS=y
CONFIG_NET_CLS_BASIC=y
CONFIG_NET_CLS_TCINDEX=y
CONFIG_NET_CLS_FW=y
CONFIG_NET_CLS_U32=y
CONFIG_NET_CLS_ACT=y
CONFIG_NET_ACT_POLICE=y
CONFIG_NET_SCH_FIFO=y
```

5.4 QT

5.4.1 Running applications

To run QT applications on the target, programs need the `-qws` parameter (i.e., run in local mode).

5.5 Serial interfaces

5.5.1 RS485

A RS485 serial interface is available on the GEAM6425 board. The device connected to this interface is `/dev/ttymxc2`. Just use common userspace system calls (e.g., `open()`, `read()`, `write()`) to access the device.

5.6 Busybox-specific commands

This Section contains a list of commands available only on the "minimal" distribution (the one containing Busybox). They are different in the full Debian distribution.

5.6.1 Boot programs

On the target, the programs started at boot are located in the `/etc/init.d/` directory. The directory contains a set of scripts. Scripts that starts with a `S` are executed during the boot. Script that starts with a `K` are executed at shutdown.

5.6.2 Telnetd

On the device, a telnet server (i.e., telnetd, provided by Busybox) is running.

To access the device through telnet, however, an account must be created using the serial interface. For security reasons, in fact, it is not allowed to access the device through telnet using the existing `root` account.

Usually, the new account is called `anonymous`. See 5.2 for information about creating a new account.

5.6.3 IP address

On the board, a DHCP client is started at boot time.

If you prefer to have a static IP address, follow the next steps:

1. Connect to the board through a serial cable
2. Open the `etc/init.d/S11-network` script
3. Comment the line containing `udhcpc`
4. Set the static IP address using the `ifconfig` command (see the example in the script itself)
5. Set the gateway using the `route` command (see the example in the script itself)

6 Technical support

A mailing list exists to join the users community or to ask technical questions about the Linux distribution.

To join the mailing list, visit the following URL:

http://ml.tux-embedded.com/mailman/listinfo/evelin-gea

The URL also contains a useful public archive of the messages sent on the mailing list.

Important: please, remember that the official language of the list is English, so do not post messages in different languages.

Bibliography

[1] Busybox, The Swiss Army Knife of Embedded Linux. http://www.busybox.net/.

[2] Das U-boot — The Universal Bootloader. http://sourceforge.net/projects/u-boot.

[3] Evidence Srl. *Evelin SDK, User's manual.*